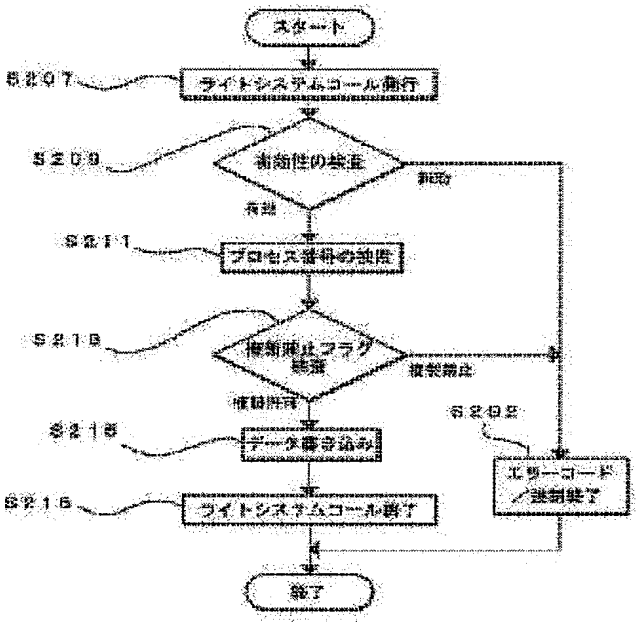
# FILE MANAGING METHOD AND FILE MANAGING PROGRAM RECORDING MEDIUM

**Publication number:** JP2001202279 (A)
**Publication date:** 2001-07-27
**Inventor(s):** SHISHIDO ICHIRO
**Applicant(s):** VICTOR COMPANY OF JAPAN
**Classification:**
**- international:** ***G06F12/14; G06F12/00; G06F21/24;*** **G06F12/14; G06F12/00; G06F21/00;** (IPC1-7): G06F12/00; G06F12/14
**- European:**
**Application number:** JP20000010733 20000119
**Priority number(s):** JP20000010733 20000119

Abstract of **JP 2001202279 (A)**
PROBLEM TO BE SOLVED: To provide a file managing method for prohibiting copying of a file while permitting the reading of the file and a file managing program recording medium. SOLUTION: An attribute on whether copying of the file is prohibited is recorded in a file attribute storing means, and the identification number of the file used by the process of an application program is recorded in a file use managing table. When some process requests writing of the file, whether the process requesting writing uses a file having the attribute prohibiting copying (S213) and when the file having the attribute prohibiting copying is used, the writing processing operation of the file is not performed.




Data supplied from the ***esp@cenet*** database — Worldwide

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－202279
（P2001－202279A）
(43)公開日　平成13年７月27日(2001.7.27)

| (51)Int.Cl.⁷ | | 識別記号 | FI | | テーマコード*(参考) |
|---|---|---|---|---|---|
| Ｇ０６Ｆ | 12/00 | ５３７ | Ｃ０６Ｆ 12/00 | ５３７Ｍ | ５Ｂ０１７ |
| | 12/14 | ３２０ | 12/14 | ３２０Ｅ | ５Ｂ０８２ |

審査請求　未請求　請求項の数３　ＯＬ　（全８頁）

(21)出願番号　特願2000－10733(P2000－10733)

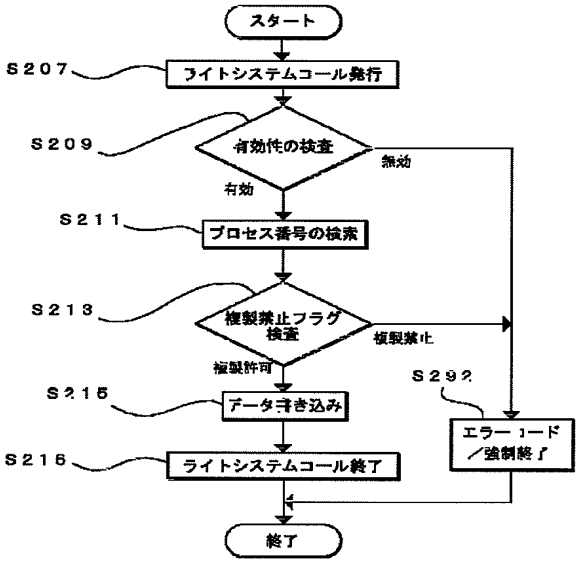
(22)出願日　平成12年１月19日(2000.1.19)

(71)出願人　000004329
日本ビクター株式会社
神奈川県横浜市神奈川区守屋町３丁目12番地
(72)発明者　宍戸　一郎
神奈川県横浜市神奈川区守屋町３丁目12番地　日本ビクター株式会社内
(74)代理人　100093067
弁理士　二瓶　正敬

Ｆターム(参考)　5B017 AA06 BA04 BB06 CA16
5B082 EA07 EA11 GA11

(54)【発明の名称】　ファイル管理方法及びファイル管理プログラム記録媒体

(57)【要約】
【課題】　ファイルの読み出しを許可するが、ファイルの複製を禁止するファイル管理方法及びファイル管理プログラム記録媒体を提供する。
【解決手段】　ファイルの複製が禁止されているか否かの属性をファイル属性格納手段に記録し、アプリケーションプログラムのプロセスにより利用された前記ファイルの識別番号を、ファイル利用管理テーブルに記録する。あるプロセスが、ファイルの書き込みを要求した場合、前記ファイル利用管理テーブルを使って、書き込み要求したプロセスが複製禁止の属性を有するファイルを利用しているかどうかを検査し（Ｓ２１３）、複製禁止の属性を有するファイルが利用されていればファイルの書き込み処理動作を行わない。



【特許請求の範囲】
【請求項１】　プロセス管理を行うオペレーティングシステムにおけるファイル管理方法であって、
ファイルの複製が禁止されているか否かを示す属性をファイルシステムのファイル属性格納手段に記録するステップと、
プロセスが前記ファイルを使用する際に、前記ファイルの属性と、前記ファイルの操作を示すフラグと、前記ファイルを利用する前記プロセスのプロセス番号とを関連付けて前記ファイル利用管理テーブルに記録するステップと、
ファイルへの書き込みを行う書き込みプロセスが前記ファイルへの書き込みを要求した場合に、前記書き込みプロセスのプロセス番号がすでに前記ファイル利用管理テーブルに記録されているか否かを判定し、前記書き込みプロセスのプロセス番号がすでに前記ファイル利用管理テーブルに記録されていた場合、前記書き込みプロセスが複製禁止の属性を有するファイルを読み出しているか否かを、前記ファイル利用管理テーブルを参照して判定するステップと、
前記書き込みプロセスが前記複製禁止の属性を有するファイルを読み出している場合、前記書き込みプロセスの前記ファイルへの書き込みを実行しないように制御するステップとを、
有するファイル管理方法。
【請求項２】　プロセス管理を行うオペレーティングシステムにおけるファイル管理方法であって、
ファイルの複製が禁止されているか否かを示す属性をファイルシステムのファイル属性格納手段に記録するステップと、
プロセスが前記ファイルを使用する際に、前記ファイルの属性と、前記ファイルの操作を示すフラグと、前記ファイルを利用する前記プロセスのプロセス番号とを関連付けてファイル利用管理テーブルに記録するステップと、
ファイルへの書き込みを行う書き込みプロセスが前記ファイルへの書き込みを要求した場合に、前記書き込みプロセスとメモリ領域を共有し得るメモリ共有プロセスのプロセス番号がすでに前記ファイル利用管理テーブルに記録されているかを判断し、前記メモリ共有プロセスのプロセス番号がすでに前記ファイル利用管理テーブルに登録されていた場合、前記メモリ共有プロセスが複製禁止の属性を有するファイルを読み出しているか否かを、前記ファイル利用管理テーブルを参照して判定するステップと、
前記メモリ共有プロセスが前記複製禁止の属性を有するファイルを読み出している場合、前記書き込みプロセスの前記ファイルへの書き込みを実行しないように制御するステップとを、
有するファイル管理方法。
【請求項３】　請求項１又は２記載のファイル管理方法を実施するプログラムが記録されたファイル管理プログラム記録媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明はファイル管理方法及びファイル管理プログラム記録媒体に関し、特にファイルシステムを利用するものに関する。
【０００２】
【従来の技術】ＵＮＩＸなどの一般的なオペレーティングシステムにおいて、ファイルを読み書きすることができる権限を特定のユーザや特定のグループにのみ与え、ファイル内容の機密性を高めることが広く行われている。従来のオペレーティングシステムにおいては、ファイルの読み出しを可能にしてファイルの書き込みを禁止する設定を行うことが可能である。
【０００３】
【発明が解決しようとする課題】一般的なオペレーティングシステムにおいて、ファイルの読み出しを許可した場合、ファイルを読み出したユーザは、そのユーザが書き込み権限を持つディレクトリにおいてそのファイルを複製することができ、ファイルの読み出し権限を有するユーザは、そのファイルの複製を制限なしに行うことができる。一方、ファイルの複製を禁止するためには、ファイルへのアクセスを全面的に禁止するしかない。このように、従来のオペレーティングシステムにおいては、あるファイルの読み出しを許可するがそのファイルの複製を禁止するという手段は提供されていない。したがって、ファイルの内容は確認でき、かつ、ファイルの２次利用を制限するという設定を行うことは不可能である。
【０００４】別な方法として、ファイルの読み出し権限を特定のアプリケーションプログラムに限定する方法がある。これは、ある特定のアプリケーションプログラムを用いなければ、ファイルにアクセスすることができないように制限する方法である。その特定のアプリケーションプログラムがファイルの複製機能を有さなければ、ファイルの複製は行えず、ファイルの複製を禁止することが可能となる。しかし、この方法ではテキストファイルや画像ファイルなど使用するファイルの種類に対応するアプリケーションプログラムをいくつも作成しなければならず、多大な手間がかかる。また、ファイルにアクセスするためには特定のアプリケーションプログラムの使用を強制され、自分でアプリケーションプログラムを開発したり、自分の使い慣れたアプリケーションプログラムを使用したりすることはできないので、ユーザにとっては非常に不都合である。
【０００５】本発明は、特定のアプリケーションプログラムの使用をユーザに強制することなく、ファイルの読み出しを許可し、かつ複製を禁止する機能をファイルシステムとして導入し、ファイル管理を行うファイル管理

方法及びファイル管理プログラム記録媒体を提供することを目的としている。

【0006】

【課題を解決するための手段】上記目的を達成するために、本発明ではファイルの複製が禁止されているか否かの属性をファイル利用管理テーブルに記録し、アプリケーションプログラムのプロセスにより利用された前記ファイルの識別番号を、ファイル利用管理テーブルに記録する。あるプロセスがファイルの書き込みを要求した場合、オペレーティングシステムは前記ファイル利用管理テーブルを検査し、ファイルの書き込み要求をしたプロセスが複製禁止の属性を有するファイルを利用している場合には、ファイルの書き込みの動作を行わない。

【0007】すなわち本発明によれば、プロセス管理を行うオペレーティングシステムにおけるファイル管理方法であって、ファイルの複製が禁止されているか否かを示す属性をファイルシステムのファイル属性格納手段に記録するステップと、プロセスが前記ファイルを使用する際に、前記ファイルの属性と、前記ファイルの操作を示すフラグと、前記ファイルを利用する前記プロセスのプロセス番号とを関連付けて前記ファイル利用管理テーブルに記録するステップと、ファイルへの書き込みを行う書き込みプロセスが前記ファイルへの書き込みを要求した場合に、前記書き込みプロセスのプロセス番号がすでに前記ファイル利用管理テーブルに記録されているか否かを判定し、前記書き込みプロセスのプロセス番号がすでに前記ファイル利用管理テーブルに記録されていた場合、前記書き込みプロセスが複製禁止の属性を有するファイルを読み出しているか否かを、前記ファイル利用管理テーブルを参照して判定するステップと、前記書き込みプロセスが前記複製禁止の属性を有するファイルを読み出している場合、前記書き込みプロセスの前記ファイルへの書き込みを実行しないように制御するステップとを、有するファイル管理方法が提供される。

【0008】また本発明によれば、プロセス管理を行うオペレーティングシステムにおけるファイル管理方法であって、ファイルの複製が禁止されているか否かを示す属性をファイルシステムのファイル属性格納手段に記録するステップと、プロセスが前記ファイルを使用する際に、前記ファイルの属性と、前記ファイルの操作を示すフラグと、前記ファイルを利用する前記プロセスのプロセス番号とを関連付けてファイル利用管理テーブルに記録するステップと、ファイルへの書き込みを行う書き込みプロセスが前記ファイルへの書き込みを要求した場合に、前記書き込みプロセスとメモリ領域を共有し得るメモリ共有プロセスのプロセス番号がすでに前記ファイル利用管理テーブルに記録されているかを判断し、前記メモリ共有プロセスのプロセス番号がすでに前記ファイル利用管理テーブルに登録されていた場合、前記メモリ共有プロセスが複製禁止の属性を有するファイルを読み出しているか否かを、前記ファイル利用管理テーブルを参照して判定するステップと、前記メモリ共有プロセスが前記複製禁止の属性を有するファイルを読み出している場合、前記書き込みプロセスの前記ファイルへの書き込みを実行しないように制御するステップとを、有するファイル管理方法が提供される。

【0009】さらに本発明によれば、上記発明のファイル管理方法を実施するプログラムが記録されたファイル管理プログラム記録媒体が提供される。

【0010】

【発明の実施の形態】以下図面を参照して本発明のファイル管理方法に係る実施の形態を説明する。図6は、本発明の全体の構成を示す模式図である。ＵＮＩＸオペレーティングシステムにおけるソフトウェアの階層と同様に、ユーザレベルとシステムレベルの２つの階層に大きく分かれている。ユーザレベルにあるアプリケーションプログラム１０がファイルの読み書きなどオペレーションシステムの機能を利用する場合には、必ずシステムコールを発行する。全てのアプリケーションプログラム１０には、一意的に定まるプロセス番号が付与され管理されている。またシステムを利用する全てのユーザには、一意的なユーザ番号が付与され管理されている。システムコールインターフェイス１２では、アプリケーションプログラム１０から発行されたシステムコールを受信及び解析する。システムコールがファイル操作に関係する場合、ファイルシステム１４にそのシステムコールの指令を伝える。このように、本発明の全体の構成は従来の構成と同一である。

【0011】ファイル操作に関する基本的なシステムコールとしては、オープン（Open）、リード（Read）、ライト（Write）、クローズ（Close）の４つがある。オペレーティングシステムは、ファイルシステム１４やデバイスドライバ１６を有しており、記憶手段１８などのハードウェアに記録されているファイルのファイル操作を行う。ファイルシステム１４は、ファイルをハードウェアに依存しない形で管理している。デバイスドライバ１６は、ＨＤＤやＣＤ－ＲＯＭドライブなどの記憶手段１８を制御する。

【0012】図7は、ファイルシステムの構成を示す模式図である。ファイルシステム１４は、パス名変換手段２０、ファイル属性格納手段２２、ファイル実体格納手段２４、ファイル利用管理テーブル２６で構成されている。パス名変換手段２０は、システムコールが指定したファイルのパス名を、ファイルを一意的に識別するファイル識別番号に変換する。ファイル属性格納手段２２には、各ファイルについてファイルの属性が格納されている。ファイル実体格納手段２４には、ファイルが実際に記録されている記録手段上の位置などが格納されている。ファイル利用管理テーブル２６には、ファイルの利用状況を管理する情報が格納される。

【００１３】図８は、ファイル属性格納手段のフィールドの構成を示す模式図である。各ファイルについて、ファイル識別番号フィールド、所有者フィールド、所有者権限フィールド、所有者以外の権限フィールド、実体の位置とサイズフィールドなどに、ファイルの属性が記録される。なお、図８ではフィールドという言葉を略しており、例えば所有者フィールドを単に所有者と表している。ファイル識別番号フィールドには、ファイルを識別する番号が記録される。所有者フィールドには、ファイルを所有するユーザのユーザ番号が記録される。所有者フィールドに記録されるファイルの所有者のみが、ファイル属性を設定することができるようになっている。

【００１４】所有者権限フィールドは、所有者が行えるファイル操作を表わしており、後で説明するように読み出しフラグ、書き込みフラグ、複製禁止フラグが記録される。所有者以外の権限フィールドは、ファイル所有者以外のユーザの権限に関して、所有者権限フィールドと同様な形式で表わしたものである。なお、本実施形態ではユーザを所有者と所有者以外の２つに大まかに分けて管理しているが、さらに細かく分類したり、複数のユーザをグループにまとめたりして管理してもよい。実体の位置とサイズフィールドは、ファイルの実体が格納されている領域の位置とサイズを表している。また、その他の属性としてはファイルの作成日時や更新日時などが記録される。

【００１５】図９は、所有者権限フィールド又は所有者以外の権限フィールドの構成を示す模式図である。所有者権限フィールド又は所有者以外の権限フィールドは同一の構成を有しており、読み出しフラグ、書き込みフラグ、複製禁止フラグが記録される。読み出しフラグが「１」の場合にはファイルを読み出すことができ、「０」の場合には読み出すことができない。書き込みフラグについても同様に、書き込みフラグが「１」の場合にはファイルを書き込むことができ、「０」の場合には書き込むことができない。また複製禁止フラグが「１」の場合にはファイルの複製は禁止され、「０」の場合はファイルの複製が許可される。

【００１６】図１０は、ファイル利用管理テーブルの構成を示す模式図である。ファイル利用管理テーブル２６の各エントリには、プロセス番号、ファイル識別番号、読み出しフラグ、書き込みフラグ、複製禁止フラグ、有効フラグが記録される。プロセス番号には、ファイルを利用するアプリケーションプログラム１０のプロセスの番号が記録される。ファイル識別番号及び複製禁止フラグには、図８に示したファイルの属性の複製禁止フラグと同じ値がセットされる。また、読み出し時には読み出しフラグが、書き込み時には書き込みフラグが各々「１」にセットされる。また、有効フラグはファイル利用管理テーブル２６そのものの有効性を示すもので「１」にセットされている場合、各フラグは有効であることを示し、「０」にセットされている場合は、読み出し又は書き込みの要求は無効となる。このファイル利用管理テーブル２６を参照すれば、そのファイルに関する各フラグのほか、そのファイルを利用しているプロセスのプロセス番号を知ることができる。

【００１７】図１１は、ファイルの読み出しを行う場合のアプリケーションプログラムの動作を説明するためのフローチャートである。図１１に示すオープンシステムコール、リードシステムコール、クローズシステムコールは、それぞれ図３に示すオープンシステムコールのフローチャート、図４に示すリードシステムコールのフローチャート、図５に示すクローズシステムコールのフローチャートを示すものである。読み出されるファイルが複製禁止か否かに関わらず、ファイルの読み出しは可能である。したがって、図１１に示すファイル読み出しの過程は、従来のファイル読み出しの過程と同一である。

【００１８】まず、ファイル読み出しに係るアプリケーションプログラム及びオペレーティングシステムの一連の動作を説明する。図３は、リードシステムコールを発行する前段階において、オープンシステムコールを発行する場合のオペレーティングシステムの動作を説明するためのフローチャートである。ステップＳ１０１において、アプリケーションプログラム１０はオープンシステムコールを発行する。このときのオープンシステムコールの操作種別は“リード”であり、オープンシステムコールの引数は使用するファイルのパス名である。ステップＳ１０３において、ファイルシステム１４はパス名の確認及びファイル操作に対する権限の確認をする。

【００１９】これは、以下の手順に従って行われる。まず、受け取ったパス名と新規作成を許可しない指示をパス名変換手段２０に与え、ファイル識別番号を得る。この場合、パス名に相当するファイルが存在しない場合には、ステップＳ１９１において、アプリケーションプログラム１０はシステムコールの戻り値としてエラーコードを受け、ファイルの読み出し処理を強制終了する。次に、パス名に相当するファイルが存在する場合には、アプリケーションプログラム１０を実行しているユーザのユーザ番号を取得する。そのユーザ番号がファイル所有者と一致する場合は、所有者の権限フィールドにおいて、読み込みフラグが「１」になっているか確認する。アプリケーションプログラム１０を実行しているユーザがファイルの所有者と異なる場合は、所有者以外の権限フィールドにおいて、読み込みフラグが「１」になっているか確認する。ユーザにファイルの読み出し権限が無い場合は、ステップＳ１９１において、アプリケーションプログラム１０はシステムコールの戻り値としてエラーコードを受け、ファイルの読み出し処理を強制終了する。パス名の正当性とユーザの権限が確認できた場合は、ステップＳ１０５において、ファイルシステム１４は有効フラグを「１」として、ファイル利用管理テーブ

ル２６に新しいエントリを作成する。ステップＳ１０６において、アプリケーションファイル１０はファイル識別番号を受け取り、オープンシステムコールを終了する。

【００２０】図４は、リードシステムコールを発行する場合のオペレーティングシステムの動作を説明するためのフローチャートである。ステップＳ１０７において、アプリケーションプログラム１０はリードシステムコールを発行する。リードシステムコールは、ファイル識別番号、データを読み込むメモリのアドレス、読み込むバイト数を引数にとる。ステップＳ１０９において、ファイルシステム１４はファイルの有効性を検査する。この検査は、アプリケーションプログラム１０のプロセス番号と指定されたファイル識別番号がファイル利用管理テーブル２６に登録されているかどうか、該当するファイルのエントリの読み出しフラグと有効フラグがともに「１」にセットされているかを調べるものである。両方とも「１」にセットされている場合は、ステップＳ１１１において、ファイルシステム１４は指定されたファイルのデータを指定されたメモリアドレスに読み出し、ステップＳ１１２において、リードシステムコールを終了する。一方、該当するエントリの読み出しフラグと有効フラグがともに「１」にセットされていないことがステップＳ１０９の検査で判明した場合は、ステップＳ１９２において、ファイルシステム１４はシステムコールインターフェイス１２を介してアプリケーションプログラム１０にエラーコードを返し、ファイルの読み出し処理を強制終了する。

【００２１】図５は、クローズシステムコールを発行する場合のオペレーティングシステムの動作を説明するためのフローチャートである。アプリケーションプログラム１０がファイルの使用を終了する場合、ステップＳ３０１において、アプリケーションプログラム１０はファイル識別番号を引数としてクローズシステムコールを発行する。ステップＳ３０３において、ファイルシステム１４は、例えばステップＳ１０５やステップＳ２０５で作成されるようなファイル利用管理テーブル２６のエントリの有効フラグを「０」に更新する。ステップＳ３０５においてクローズシステムコールを終了する。なお、クローズシステムコールが発行された段階では、ファイル利用管理テーブル２６のエントリは削除されない。該当プロセス及び該当プロセスとメモリを共有する可能性のあるプロセスが全て終了した後に削除される。本明細書では、あるプロセスとメモリを共有するプロセスのことをメモリ共有プロセスと呼ぶこともある。

【００２２】図１２は、ファイルの書き込みを行う場合のアプリケーションプログラムの動作を説明するためのフローチャートである。図１２に示すオープンシステムコール、ライトシステムコール、クローズシステムコールは、それぞれ図２に示すオープンシステムコールのフローチャート、図１に示すライトシステムコールのフローチャート、図５に示すクローズシステムコールのフローチャートを示すものである。複製許可されているファイルをユーザが複製しようとする場合には、図１２に示すフローチャートに従って、あるファイルを新たなファイルに書き込むことが可能である。これにより、あるファイルを複製して別の新しいファイルを作成することが可能である。

【００２３】次に、ファイル書き込みに係るアプリケーションプログラム及びオペレーティングシステムの一連の動作を説明する。図２は、ライトシステムコールを発行する前段階において、オープンシステムコールを発行する場合のオペレーティングシステムの動作を説明するためのフローチャートである。ステップＳ２０１において、アプリケーションプログラム１０はオープンシステムコールを発行する。このときのオープンシステムコールの操作種別は“ライト”である。ステップＳ２０３において、ファイルシステム１４はパス名の確認及びファイル操作に対する権限の確認をする。ステップＳ２０３は、以下の手順に従って行われる。まず、受け取ったパス名と新規作成を許可する指示をパス名変換手段２０に与え、ファイル識別番号を得る。この場合、パス名に相当するファイルが存在しない場合には、パス名変換手段２０により新規のファイル識別番号が割り当てられる。既存ファイルを書き換える場合には、ファイル読み出しの場合と同様に、そのファイルの「所有者権限」あるいは「所有者以外の権限」の書き込みフラグを検査し、そのユーザが書き込み権限を持っていることを確認する。ユーザにファイルの書き込み権限が無い場合は、ステップＳ２９１において、アプリケーションプログラム１０はシステムコールの戻り値としてエラーコードを受け、ファイルの書き込み処理を強制終了する。パス名の正当性とユーザの権限が確認できた場合は、ステップＳ２０５において、ファイルシステム１４はファイル利用管理テーブル２６に有効フラグを「１」として新しいエントリを作成する。Ｓ２０６において、アプリケーションファイル１０はファイル識別番号を受け取り、オープンシステムコールを終了する。

【００２４】図１は、ライトシステムコールを発行する場合のオペレーティングシステムの動作を説明するためのフローチャートである。ステップＳ２０７において、アプリケーションプログラム１０はライトシステムコールを発行する。ライトシステムコールは、ファイル識別番号、書き込むデータを保持しているメモリアドレス、書き込むバイト数を引数に取る。ステップＳ２０９において、ファイルシステム１４はファイルの有効性を検査する。これは、アプリケーションプログラム１０のプロセス番号と指定されたファイル識別番号がファイル利用管理テーブル２６に登録されているかどうか、該当するエントリの書き込みフラグと有効フラグがともに「１」

にセットされているかを検査するものである。これらのプロセス番号やファイル番号がファイル利用管理テーブル２６に登録されていない場合、または書き込みフラグと有効フラグがともに「１」にセットされていない場合は、ステップＳ２９２において、ファイルシステム１４はシステムコールインターフェイス１２を介してアプリケーションプログラム１０にエラーコードを返し、ファイルの書き込み処理を強制終了する。

【００２５】次に、ステップＳ２１１において、オペレーティングシステムはライトシステムコールを発行したプロセス及びそのプロセスとメモリ領域を共有する可能性のあるプロセスのプロセス番号を検索する。メモリ領域を共有する可能性があるメモリ共有プロセスとしては、例えばライトシステムコールを発行したプロセスの親プロセス及び子プロセスが挙げられる。プロセス間の親子関係は、オペレーティングシステムが管理するプロセステーブルを調べることにより分かる。

【００２６】ステップＳ２１３において、ファイルシステム１４は、ファイル利用管理テーブル２６のファイル利用管理テーブルのエントリの複製禁止フラグを検査する。この場合、検査するエントリは、プロセス番号がライトシステムコールを発行したプロセスのプロセス番号に一致するエントリ、あるいはメモリを共有する可能性があるプロセスのプロセス番号に一致するエントリである。そのようなエントリが存在し、該当エントリの読み出しフラグが「１」で、なおかつ複製禁止フラグが「１」である場合には、書き込み処理を行わず、ステップＳ２９２において、エラーコードをアプリケーションプログラムに返し、ファイルの書き込み処理を強制終了する。そのような条件に該当しなければ、ステップＳ２１５において、指定されたメモリアドレスのデータを新たなファイルに書き込んで、ステップＳ２１６において、ライトシステムコールを終了する。

【００２７】一方、ユーザが複製禁止のファイルを複製しようとする場合を以下に説明する。なお、アプリケーションプログラム１０を実行するユーザは、対象としているファイルの読み出し権限を持つが、複製する権限は持たないものとする。図１１に示すフローチャートに従って、アプリケーションプログラムは複製禁止のファイルの読み出しを行う。ファイルの読み出しは、ファイルの複製が禁止されているか否かに関わらず行われるので、ファイルの読み出しは正常に行われる。このようにして、複製が禁止されている複製元のファイルをメモリ領域に複製データとして読み込むことができる。なお、メモリ領域に読み込まれたデータを複製データと呼ぶことにする。

【００２８】次に、図１に示すフローチャートに従って、アプリケーションプログラムはメモリ領域にある複製データを新しいファイルに書き込もうとする。しかし、複製禁止フラグを有するファイルを読み出しているので、このファイルを読み出したときの読み出しプロセスの複製禁止フラグが複製禁止であることがファイル利用管理テーブル２６に記録されている。したがって、ステップＳ２１３で、ファイルシステム１４は読み出しプロセスの複製禁止フラグが複製禁止を示す「１」であることを見つけて、メモリ領域にある複製データを新しいファイルへの書き込みを拒否する。その結果、アプリケーションはエラーコードを受け、ファイルへの書き込み処理を終了する。このようにして、ユーザは複製禁止のファイルを読み出すことは可能であるが、読み出したファイルを書き込んで新しいファイルを作成することができない。以上のように、ファイルの属性として複製を禁止するフラグを設けることによって、従来のオープン、リード、ライト、クローズの４つのシステムコールを有するファイルシステム１４において、読み出しは許可されるが複製は許可されないように設定することが可能となる。

【００２９】

【発明の効果】本発明は、ファイルの書き込みを行う場合にそのファイルの読み出されたプロセスが記録されたファイル利用管理テーブルを参照して、複製が禁止されているか否かを判断するので、ファイルの読み出しを許可し、なおかつファイルの複製を禁止することが可能になる。また、本発明は、オペレーティングシステムの一部であるファイルシステムの機能を利用しているので、ユーザに特別なアプリケーションの使用を強制する必要がない。また、特別なアプリケーションプログラムを作る必要もなく、アプリケーションプログラムの開発コストも削減される。

【図面の簡単な説明】

【図１】ライトシステムコールを発行する場合のオペレーティングシステムの動作を説明するためのフローチャートである。

【図２】ライトシステムコールを発行する前段階において、オープンシステムコールを発行する場合のオペレーティングシステムの動作を説明するためのフローチャートである。

【図３】リードシステムコールを発行する前段階において、オープンシステムコールを発行する場合のオペレーティングシステムの動作を説明するためのフローチャートである。

【図４】リードシステムコールを発行する場合のオペレーティングシステムの動作を説明するためのフローチャートである。

【図５】クローズシステムコールを発行する場合のオペレーティングシステムの動作を説明するためのフローチャートである。

【図６】本発明の全体の構成を示す模式図である。

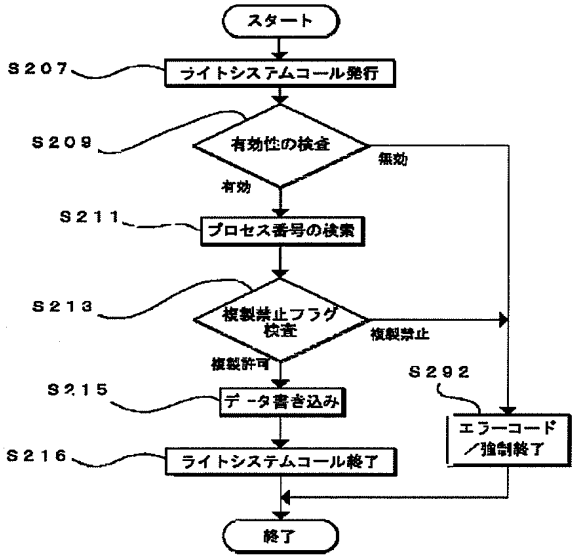
【図７】ファイルシステムの構成を示す模式図である。

【図８】ファイル属性格納手段のフィールドの構成を示

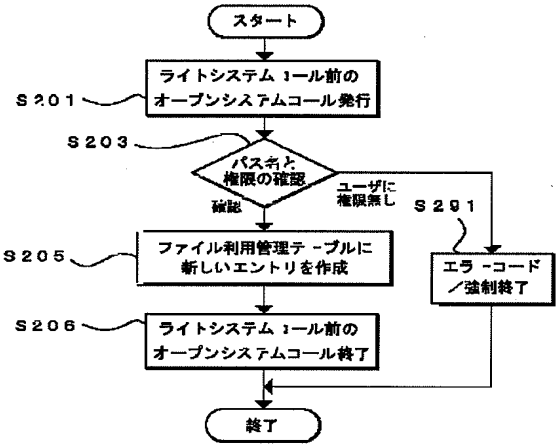
す模式図である。
【図９】所有者権限フィールド又は所有者以外の権限フィールドの構成を示す模式図である。
【図１０】ファイル利用管理テーブルの構成を示す模式図である。
【図１１】ファイルの読み出しを行う場合のアプリケーションプログラムの動作を説明するためのフローチャートである。
【図１２】ファイルの書き込みを行う場合のアプリケーションプログラムの動作を説明するためのフローチャートである。

【符号の説明】
１０　アプリケーションプログラム
１２　システムコールインターフェイス
１４　ファイルシステム
１６　デバイスドライバ
１８　記憶手段
２０　パス名変換手段
２２　ファイル属性格納手段
２４　ファイル実体格納手段
２６　ファイル利用管理テーブル

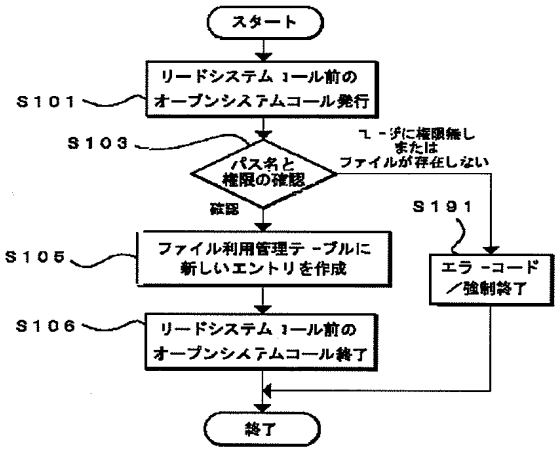
【図１】



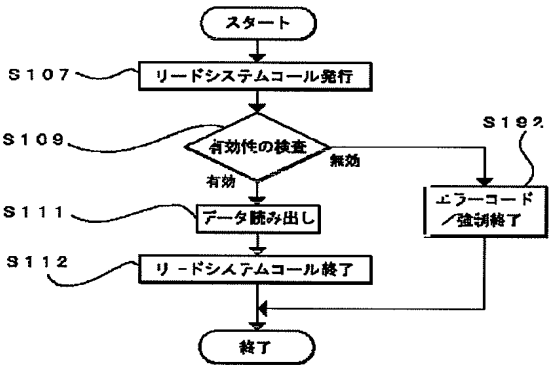

【図２】



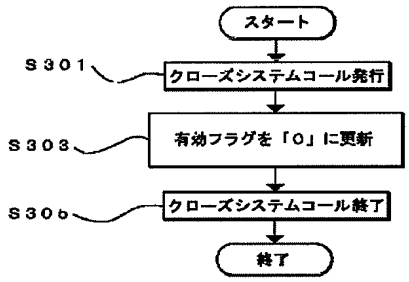

【図３】



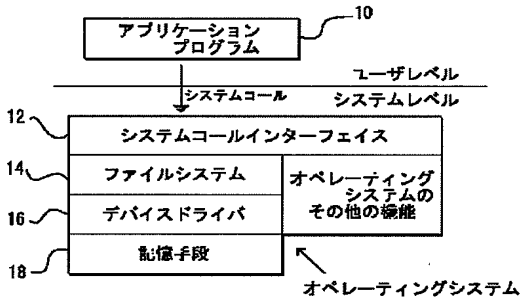

【図４】



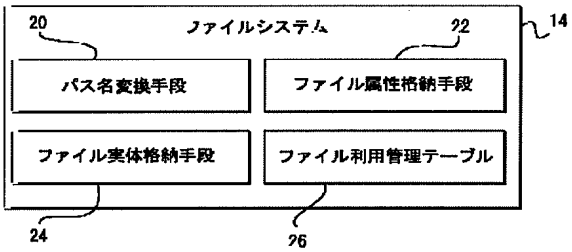
【図5】




【図6】




【図7】




【図8】

ファイル属性格納手段

| ファイル識別番号 | 所有者 | 所有者権限 | 所有者以外の権限 | 実体の位置とサイズ | その他 |
|---|---|---|---|---|---|

【図9】

所有者又は所有者以外の権限フィールドの構成

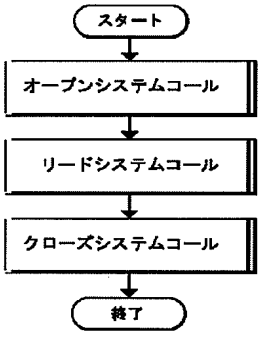
| 読み出しフラグ | 書き込みフラグ | 複製禁止フラグ |
|---|---|---|
| 0 か 1 | 0 か 1 | 0 か 1 |

【図10】

ファイル利用管理テーブル

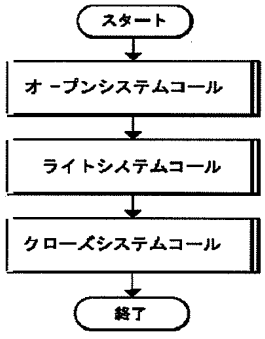
| プロセス番号 | ファイル識別番号 | 読み出しフラグ | 書き込みフラグ | 複製禁止フラグ | 有効フラグ |
|---|---|---|---|---|---|

【図11】




ファイルの読み出し

【図12】




ファイルの書き込み

* NOTICES *



1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

---

## CLAIMS

---

[Claim(s)]
[Claim 1]A file management method characterized by comprising the following in an operating system which performs process control.
A step which records an attribute which shows whether a duplicate of a file is forbidden on a file attribute storing means of a file system.
When a process uses said file, it is the attribute of said file.
A flag which shows operation of said file.
A step which associates a process number of said process of using said file, and is recorded on said file use managing table, When a write-in process of performing writing to a file requires writing to said file, When it judges whether a process number of said write-in process is already recorded on said file use managing table and a process number of said write-in process is already recorded on said file use managing table, said write-in process is the attribute of duplication prohibition.

[Claim 2]A file management method characterized by comprising the following in an operating system which performs process control.
A step which records an attribute which shows whether a duplicate of a file is forbidden on a file attribute storing means of a file system.
When a process uses said file, it is the attribute of said file.
A flag which shows operation of said file.
A step which associates a process number of said process of using said file, and is recorded on a file use managing table, When a write-in process of performing writing to a file requires writing to said file, It is judged whether a process number of a memory share process that said write-in process and a memory area can be shared is already recorded on said file use managing table, When a process number of said memory share process is already registered into said file use managing table, said memory share process is the attribute of duplication prohibition.

[Claim 3]A file management program recording medium with which a program which enforces the file management method according to claim 1 or 2 was recorded.

---

[Translation done.]

* NOTICES *



1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

## DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention]Especially this invention relates to the thing using a file system about a file management method and a file management program recording medium.
[0002]
[Description of the Prior Art]In general operating systems, such as UNIX, the power that a file can be written is lodged only in a specific user or a specific group, and improving the confidentiality of a file content is performed widely. In the conventional operating system, it is possible to perform setting out which makes read-out of a file possible and forbids the writing of a file.
[0003]
[Problem(s) to be Solved by the Invention]In a general operating system, when read-out of a file is permitted, the user who read the file, The user who can reproduce the file in the directory which the user writes in and has authority, and has the read-out authority of a file can perform reproduction of the file without restriction. On the other hand, in order to forbid the duplicate of a file, access to a file must be forbidden extensively. Thus, in the conventional operating system, although read-out of a certain file is permitted, a means to forbid the duplicate of the file is not provided. Therefore, it is impossible to perform setting out of being able to check the contents of a file and restricting secondary use of a file.
[0004]Another methods include the method of limiting the read-out authority of a file to a specific application program. If a certain specific application program is not used for this, it is the method of restricting so that a file cannot be accessed. If the specific application program does not have a copy function of a file, reproduction of a file cannot be performed but it becomes possible to forbid the duplicate of a file. However, by this method, many application programs corresponding to the kind of files to be used, such as a text file and a graphics file, must be created, and it takes great time and effort. Since use of a specific application program can be forced, and an application program cannot be developed by itself or the application program to which he is used cannot be used in order to access a file, for a user, it is dramatically inconvenient.
[0005]This invention, without forcing a user into use of a specific application program, The function to permit read-out of a file and to forbid a duplicate is introduced as a file system, and it aims at providing the file management method and file management program recording medium which perform file management.
[0006]
[Means for Solving the Problem]To achieve the above objects, in this invention, the attribute of whether a duplicate of a file is forbidden is recorded on a file use managing table, and an identification number of said file used by a process of an application program is recorded on a file use managing table. When a certain process requires writing of a file, an operating system inspects said file use managing table, and writing of a file is not operated when a file in which a process which carried out a write request of a file has the attribute of duplication prohibition is used.
[0007]Namely, according to this invention, it is a file management method in an operating system which performs process control, A step which records an attribute which shows whether a duplicate of a file is forbidden on a file attribute storing means of a file system, A flag which shows the attribute of said file, and operation of said file when a process uses said file, A step which associates a process number of said process of using said file, and is recorded on said file use managing table, When a write-in process of performing writing to a file requires writing to said file, It is judged whether a process number of said write-in process is already recorded on said file use managing table, When a process number of said write-in process is already recorded on said file use managing table, A step which judges whether said write-in

process has read a file which has the attribute of duplication prohibition with reference to said file use managing table, When said write-in process has read a file which has the attribute of said duplication prohibition, a step controlled not to perform writing to said file of said write-in process, A file management method which it has is provided.

[0008]According to this invention, it is a file management method in an operating system which performs process control, A step which records an attribute which shows whether a duplicate of a file is forbidden on a file attribute storing means of a file system, A flag which shows the attribute of said file, and operation of said file when a process uses said file, A step which associates a process number of said process of using said file, and is recorded on a file use managing table, When a write-in process of performing writing to a file requires writing to said file, It is judged whether a process number of a memory share process that said write-in process and a memory area can be shared is already recorded on said file use managing table, When a process number of said memory share process is already registered into said file use managing table, When a step which judges whether said memory share process has read a file which has the attribute of duplication prohibition with reference to said file use managing table, and a file in which said memory share process has the attribute of said duplication prohibition are read, A file management method which has a step controlled not to perform writing to said file of said write-in process is provided.

[0009]Furthermore, according to this invention, a file management program recording medium with which a program which enforces a file management method of the above-mentioned invention was recorded is provided.

[0010]

[Embodiment of the Invention]The embodiment which starts the file management method of this invention with reference to drawings below is described. Drawing 6 is a mimetic diagram showing the composition of whole this invention. It is roughly divided into two hierarchies, user levels and a system level, as well as the hierarchy of the software in a UNIX operating system. When the application program 10 in user levels uses the function of operation system, such as reading and writing of a file, a system call is certainly published. The process number which becomes settled uniquely is given to all the application programs 10, and it is managed. The unique user number is given and managed by all the users using a system. In the system call interface 12, the system call published from the application program 10 is received and analyzed. When a system call is related to a file operation, instructions of the system call are told to the file system 14. Thus, the composition of whole this invention is the same as the conventional composition.

[0011]As a fundamental system call about a file operation, there are four, open (Open) and lead (Read), light (Write), and closing (Close). The operating system has the file system 14 and the device driver 16, and performs the file operation of the file currently recorded on hardwares, such as the memory measure 18. The file system 14 is managed in the form where it does not depend for a file on hardware. The device driver 16 controls the memory measure 18 of HDD, a CD-ROM drive, etc.

[0012]Drawing 7 is a mimetic diagram showing the composition of a file system. The file system 14 comprises the pathname conversion method 20, the file attribute storing means 22, the file substance storing means 24, and the file use managing table 26. As for the pathname conversion method 20, the pathname of the file specified by a system call is changed into the file identification number which identifies a file uniquely. The attribute of a file is stored in the file attribute storing means 22 about each file. The position on the recording device on which the file is actually recorded, etc. are stored in the file substance storing means 24. The information which manages the using state of a file is stored in the file use managing table 26.

[0013]Drawing 8 is a mimetic diagram showing the composition of the field of a file attribute storing means. About each file, the attribute of a file is recorded on a file identification number field, the owner field, the owner authority field, the authority fields other than an owner, a position, a size field of substance, etc. The word "field" is omitted, for example, the owner field is only expressed in drawing 8 as the owner. The number which identifies a file is recorded on a file identification number field. The user number of the user who owns a file is recorded on the owner field. Only the owner of the file recorded on the owner field can set up a file attribute now.

[0014]The owner authority field expresses the file operation which an owner can perform, it reads it so that it may explain later, and a flag, a write-in flag, and a duplication prohibition flag are recorded. The authority fields other than an owner are expressed with the same form as the owner authority field about the authority of users other than a file owner. Although the user was divided roughly other than an owner and an owner to two and is managed in this embodiment, it may classify still more finely, or two or more users may be summarized in a group, and it may manage. The position and size field of substance express the position and size of the field in which the substance of a file is stored. As other attributes, the date and

time of creation, an update date, etc. of a file are recorded.
[0015]Drawing 9 is a mimetic diagram showing the composition of the owner authority field or the authority fields other than an owner. The owner authority field or the authority fields other than an owner have the same composition, and a read-out flag, a write-in flag, and a duplication prohibition flag are recorded. When a read-out flag is "1", a file can be read, and when it is "0", it cannot read. About a write-in flag, similarly, when a write-in flag is "1", a file can be written in, and when it is "0", it cannot write in. When a duplication prohibition flag is "1", the duplicate of a file is forbidden, and in the case of "0", the duplicate of a file is permitted.
[0016]Drawing 10 is a mimetic diagram showing the composition of a file use managing table. A process number, a file identification number, a read-out flag, a write-in flag, a duplication prohibition flag, and a valid flag are recorded on each entry of the file use managing table 26. The number of the process of the application program 10 of using a file is recorded on a process number. The same value as the duplication prohibition flag of the attribute of the file shown in drawing 8 is set to a file identification number and a duplication prohibition flag. It reads at the time of read-out, a flag writes in at the time of writing, and a flag is respectively set to "1." When it is shown that each flag is effective when a valid flag shows the validity of file use managing table 26 itself and it is set to "1" and it is set to "0", the demand of read-out or writing becomes invalid. If this file use managing table 26 is referred to, the process number of the process using its file besides each flag about that file can be known.
[0017]Drawing 11 is a flow chart for explaining operation of the application program in the case of reading a file. The open system call shown in drawing 11, a lead system call, and a closing system call, The flow chart of the open system call shown in drawing 3, respectively, the flow chart of the lead system call shown in drawing 4, and the flow chart of the closing system call shown in drawing 5 are shown. It is not concerned for the file read to be duplication prohibition, but read-out of a file is possible. Therefore, the process of file read-out shown in drawing 11 is the same as the process of file read-out of the former.
[0018]First, operation of a series of the application program and operating system concerning file read-out is explained. In the preceding paragraph story which publishes a lead system call, drawing 3 is a flow chart for explaining operation of the operating system in the case of publishing an open system call. In Step S101, the application program 10 publishes an open system call. The operation classification of the open system call at this time is "a lead", and the argument of an open system call is a pathname of the file to be used. In Step S103, the file system 14 checks the authority over a check and file operation of a pathname.
[0019]This is performed according to the following procedures. First, the received pathname and the directions with which new production is not permitted are given to the pathname conversion method 20, and a file identification number is acquired. In this case, when the file equivalent to a pathname does not exist, in Step S191, the application program 10 receives an error code as a return value of a system call, and forces the reading processing of a file to terminate. Next, when the file equivalent to a pathname exists, the user number of the user who is executing the application program 10 is acquired. When the user number is in agreement with a file owner, in an owner's authority field, it is checked whether the reading flag is "1." When the user who is executing the application program 10 differs from the owner of a file, in the authority fields other than an owner, it is checked whether the reading flag is "1." When a user does not have the read-out authority of a file, in Step S191, the application program 10 receives an error code as a return value of a system call, and forces the reading processing of a file to terminate. When the justification of a pathname and a user's authority are able to be checked, in Step S105, the file system 14 sets a valid flag to "1", and creates a new entry to the file use managing table 26. In Step S106, the application file 10 receives a file identification number, and ends an open system call.
[0020]Drawing 4 is a flow chart for explaining operation of the operating system in the case of publishing a lead system call. In Step S107, the application program 10 publishes a lead system call. A lead system call takes the address of the memory which reads a file identification number and data, and the number of bytes to read to an argument. In Step S109, the file system 14 inspects the validity of a file. This inspection investigates whether both the read-out flags and valid flags of the entry of the file which corresponds [ whether the file identification number specified as the process number of the application program 10 is registered into the file use managing table 26 and ] are set to "1." When both are set to "1", in Step S111, the file system 14 is read to the memory address which had data of the specified file specified, and ends a lead system call in Step S112. When it becomes clear by the inspection of Step S109 that neither of the read-out flag and valid flag of an applicable entry are set to "1" on the other hand, In Step S192, the file system 14 returns an error code to the application program 10 via the system call interface 12, and forces the reading processing of a file to terminate.

[0021]Drawing 5 is a flow chart for explaining operation of the operating system in the case of publishing a closing system call. When the application program 10 ends use of a file, in Step S301, the application program 10 publishes a closing system call by making a file identification number into an argument. In Step S303, the file system 14 updates to "0" the valid flag of the entry of the file use managing table 26 which is created, for example at Step S105 or Step S205. A closing system call is ended in Step S305. In the stage where the closing system call was published, the entry of the file use managing table 26 is not deleted. It is deleted after all of an applicable process and an applicable process, and the process that may share a memory are completed. In this specification, a certain process and the process of sharing a memory may be called a memory share process.
[0022]Drawing 12 is a flow chart for explaining operation of the application program in the case of writing in a file. The open system call shown in drawing 12, a Wright system call, and a closing system call, The flow chart of the open system call shown in drawing 2, respectively, the flow chart of the Wright system call shown in drawing 1, and the flow chart of the closing system call shown in drawing 5 are shown. When a user is going to reproduce the file by which duplicate permission is carried out, it is possible to write a certain file in a new file according to the flow chart shown in drawing 12. It is possible to reproduce a certain file and to create another new file by this.
[0023]Next, operation of a series of the application program and operating system concerning file writing is explained. In the preceding paragraph story which publishes a Wright system call, drawing 2 is a flow chart for explaining operation of the operating system in the case of publishing an open system call. In Step S201, the application program 10 publishes an open system call. The operation classification of the open system call at this time is a "light." In Step S203, the file system 14 checks the authority over a check and file operation of a pathname. Step S203 is performed according to the following procedures. First, the received pathname and the directions with which new production is permitted are given to the pathname conversion method 20, and a file identification number is acquired. In this case, when the file equivalent to a pathname does not exist, a new file identification number is assigned by the pathname conversion method 20. In rewriting the existing file, it checks inspecting the write-in flag of the "owner authority" of the file, or "authority other than an owner", and the user writing in like the case of file read-out, and having authority. When a user does not have the write-in authority of a file, in Step S291, the application program 10 receives an error code as a return value of a system call, and forces the writing processing of a file to terminate. When the justification of a pathname and a user's authority are able to be checked, in Step S205, the file system 14 creates an entry new as "1" for a valid flag to the file use managing table 26. In S206, the application file 10 receives a file identification number, and ends an open system call.
[0024]Drawing 1 is a flow chart for explaining operation of the operating system in the case of publishing a Wright system call. In Step S207, the application program 10 publishes a Wright system call. A Wright system call takes a file identification number, the memory address holding the data to write in, and the number of bytes to write in to an argument. In Step S209, the file system 14 inspects the validity of a file. This inspects whether both the write-in flags and valid flags of the entry which corresponds [ whether the file identification number specified as the process number of the application program 10 is registered into the file use managing table 26 and ] are set to "1." When these process numbers or file numbers are not registered into the file use managing table 26, or when neither the write-in flag nor the valid flag is set to "1", In Step S292, the file system 14 returns an error code to the application program 10 via the system call interface 12, and forces the writing processing of a file to terminate.
[0025]Next, in Step S211, an operating system searches the process number of the process which published the Wright system call and its process, and the process which may share a memory area. As a memory share process that a memory area may be shared, the parent process and child process of a process which published the Wright system call, for example are mentioned. By investigating the process table which an operating system manages shows the child-parent relationship between processes.
[0026]In Step S213, the file system 14 inspects the duplication prohibition flag of the entry of the file use managing table of the file use managing table 26. In this case, the entry to inspect is an entry whose process number corresponds with the process number of the process which published the Wright system call, or an entry which is in agreement with the process number of the process which may share a memory. Such an entry exists, and by "1", when a duplication prohibition flag is moreover "1", the read-out flag of an applicable entry does not perform writing processing, but returns an error code to an application program in Step S292, and forces the writing processing of a file to terminate. If it does not correspond to such conditions, in Step S215, the data of the specified memory address is written in a new file, and a Wright system call is ended in Step S216.
[0027]On the other hand, the case where a user is going to reproduce the file of duplication prohibition is

explained below. It shall not have the authority to reproduce, although the user who executes the application program 10 has the read-out authority of the target file. According to the flow chart shown in drawing 11, an application program reads the file of duplication prohibition. Since read-out of a file is not concerned with whether the duplicate of a file is forbidden but is performed, read-out of a file is performed normally. Thus, the file of the duplicate origin to which the duplicate is forbidden can be read into a memory area as duplication data. The data read into the memory area will be called duplication data.
[0028]Next, according to the flow chart shown in drawing 1, an application program tries to write the duplication data in a memory area in a new file. However, since the file which has a duplication prohibition flag is read, it is recorded on the file use managing table 26 that the duplication prohibition flag of a read-out process when this file is read is duplication prohibition. Therefore, it finds that the file system 14 is "1" the duplication prohibition flag of a read-out process indicates duplication prohibition to be at Step S213, and the writing to a new file is refused for the duplication data in a memory area. As a result, application receives an error code and ends the writing processing to a file. Thus, although it is possible, that a user reads the file of duplication prohibition cannot write in the read file, and it cannot create a new file. As mentioned above, by forming the flag which forbids a duplicate as an attribute of a file, in the file system 14 which has the conventional opening, a lead, a light, and four system calls of closing, although read-out is permitted, a duplicate becomes possible [ setting up so that a permission may not be granted ].
[0029]
[Effect of the Invention]It becomes possible to permit read-out of a file and to forbid the duplicate of a file moreover, since it judges whether the duplicate is forbidden or not with reference to the file use managing table in which the process from which the file was read was recorded when this invention writes in a file. Since this invention uses the function of the file system which is a part of operating system, it does not need to force use of application special to a user. It is not necessary to make a special application program, and the development cost of an application program is also reduced.

---

[Translation done.]

* NOTICES *



1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

## DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1]It is a flow chart for explaining operation of the operating system in the case of publishing a Wright system call.
[Drawing 2]In the preceding paragraph story which publishes a Wright system call, it is a flow chart for explaining operation of the operating system in the case of publishing an open system call.
[Drawing 3]In the preceding paragraph story which publishes a lead system call, it is a flow chart for explaining operation of the operating system in the case of publishing an open system call.
[Drawing 4]It is a flow chart for explaining operation of the operating system in the case of publishing a lead system call.
[Drawing 5]It is a flow chart for explaining operation of the operating system in the case of publishing a closing system call.
[Drawing 6]It is a mimetic diagram showing the composition of whole this invention.
[Drawing 7]It is a mimetic diagram showing the composition of a file system.
[Drawing 8]It is a mimetic diagram showing the composition of the field of a file attribute storing means.
[Drawing 9]It is a mimetic diagram showing the composition of the owner authority field or the authority fields other than an owner.
[Drawing 10]It is a mimetic diagram showing the composition of a file use managing table.
[Drawing 11]It is a flow chart for explaining operation of the application program in the case of reading a file.
[Drawing 12]It is a flow chart for explaining operation of the application program in the case of writing in a file.
[Description of Notations]
10 Application program
12 System call interface
14 File system
16 Device driver
18 Memory measure
20 Pathname conversion method
22 File attribute storing means
24 File substance storing means
26 File use managing table

[Translation done.]

* NOTICES *

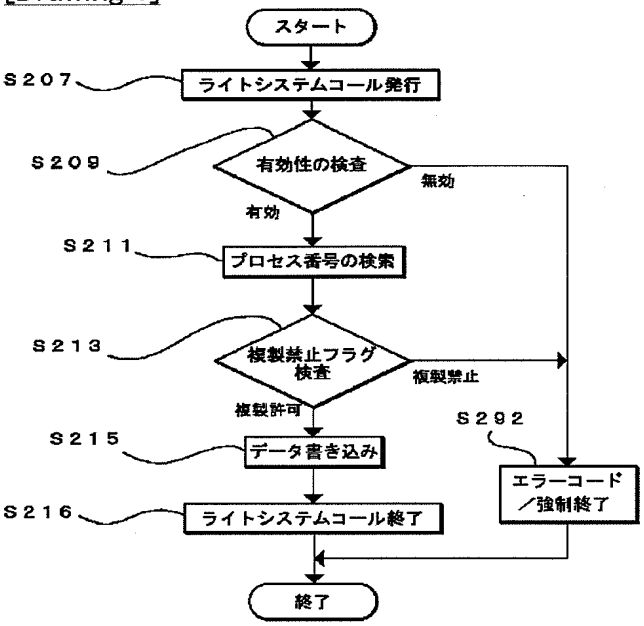


1.This document has been translated by computer. So the translation may not reflect the original precisely.

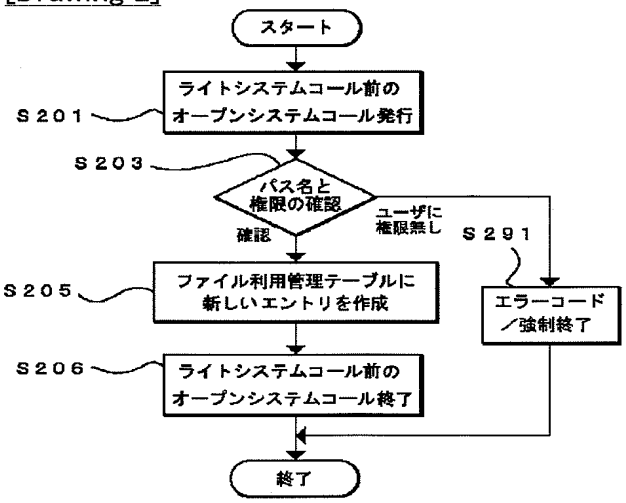
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

## DRAWINGS

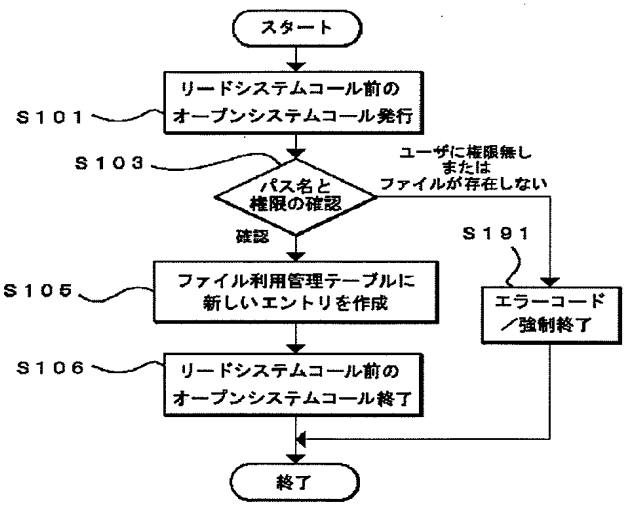
[Drawing 1]



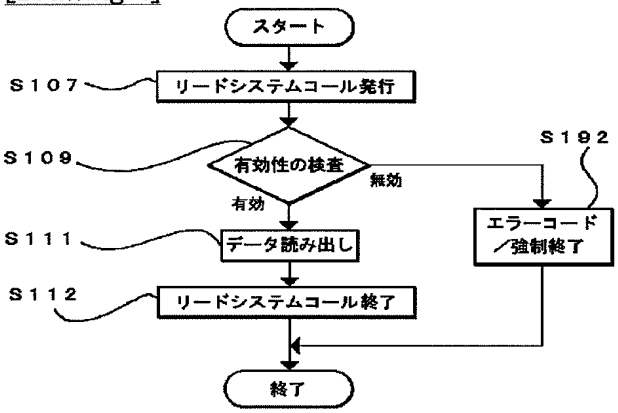

[Drawing 2]



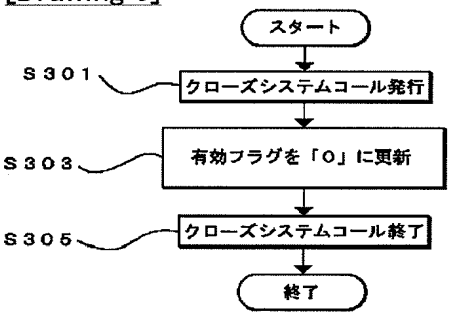

[Drawing 3]



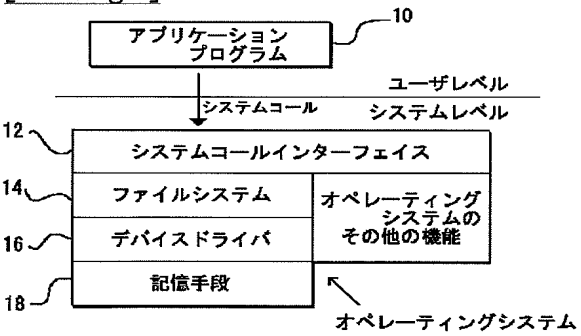

[Drawing 4]



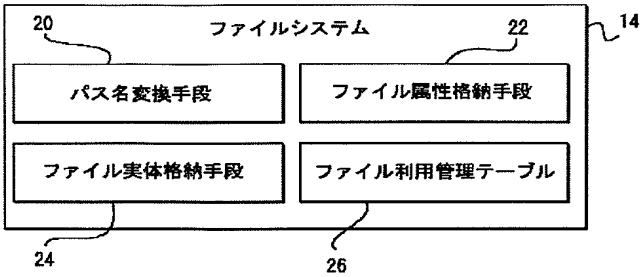

[Drawing 5]




[Drawing 6]




[Drawing 7]




[Drawing 8]
ファイル属性格納手段

| ファイル識別番号 | 所有者 | 所有者権限 | 所有者以外の権限 | 実体の位置とサイズ | その他 |
|---|---|---|---|---|---|

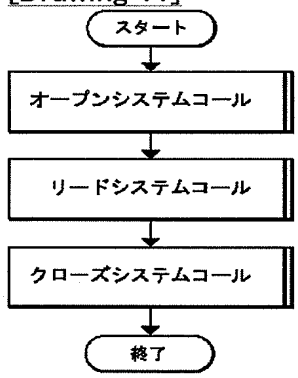
[Drawing 9]
所有者又は所有者以外の権限フィールドの構成

| 読み出しフラグ 0 か 1 | 書き込みフラグ 0 か 1 | 複製禁止フラグ 0 か 1 |
|---|---|---|

[Drawing 10]
ファイル利用管理テーブル

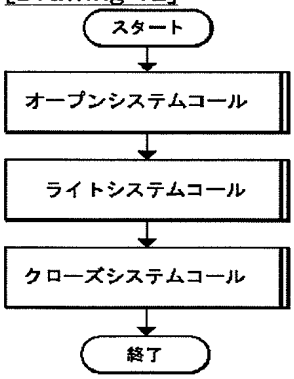
| プロセス番号 | ファイル識別番号 | 読み出しフラグ | 書き込みフラグ | 複製禁止フラグ | 有効フラグ |
|---|---|---|---|---|---|

[Drawing 11]




ファイルの読み出し

[Drawing 12]




ファイルの書き込み

---

[Translation done.]